Hepatoprotective effects of systemic ER activation

BulkRNAseq - Transcriptome molecular signatures

Christian Sommerauer & Carlos Gallardo

25 July, 2023

```
# source and library import
source('code/00_helper_functions.R')
library(tidyverse)
library(DESeq2)
library(ggvenn)
library(clusterProfiler)
library(org.Mm.eg.db)
library(ggrepel)
```

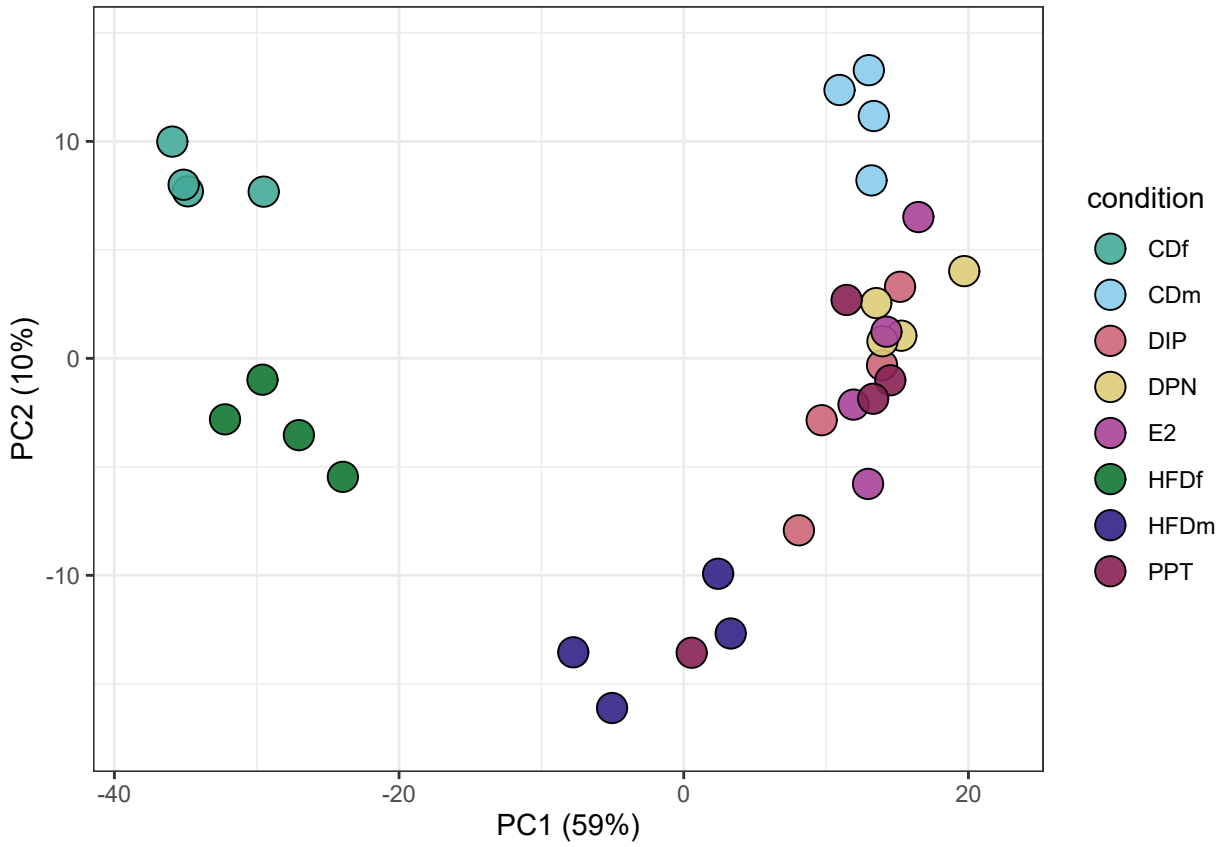
```
# color palettes
colPals <- list()
colPals$conditions <- setNames(c('#44AA99', '#117733', '#88CCEE', '#332288', '#DDCC77', '#CC6677', '#AA4499', '#882255'),
                               c('CDf', 'HFDf', 'CDm', 'HFDm', 'DPN', 'DIP', 'E2', 'PPT'))
colPals$RdBu <- rev(RColorBrewer::brewer.pal(n=11, name = 'RdBu'))
colPals$UpDown <- setNames(colPals$RdBu[c(10,2)],
                           c('up', 'down'))
```

Load data

```
# consensus differentially expressed genes
DEGs <- readRDS('results/bulkRNAseq_mmus_DEGs.rds')

# raw counts RNAseq
raw_counts <- read.table(
  file = 'data/bulkRNAseq_mmus_rawcounts.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE) %>%
  dplyr::filter(geneID %in% DEGs$unfilt$CDfVsCDm$ensembl_gene_id) %>%
  tibble::column_to_rownames('geneID') %>%
  as.matrix()

# gene lengths
gene_len <- read.table(
  file = 'data/bulkRNAseq_mmus_gene_lengths.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE) %>%
  dplyr::filter(geneID %in% DEGs$unfilt$CDfVsCDm$ensembl_gene_id)

# design RNAseq
design_meta <- read.table(
  file = 'data/bulkRNAseq_mmus_design.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE)

# ensembl gene annotation (Mus musculus)
gene_ann <- read.table(
  file = 'data/ensembl_mmus_sep2019_annotation.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE,
  fill = FALSE,
  quote = '') %>%
  dplyr::filter(ensembl_gene_id %in% DEGs$unfilt$CDfVsCDm$ensembl_gene_id) %>%
  dplyr::arrange(factor(ensembl_gene_id, levels = rownames(raw_counts)))
```

Principal component analysis (PCA)

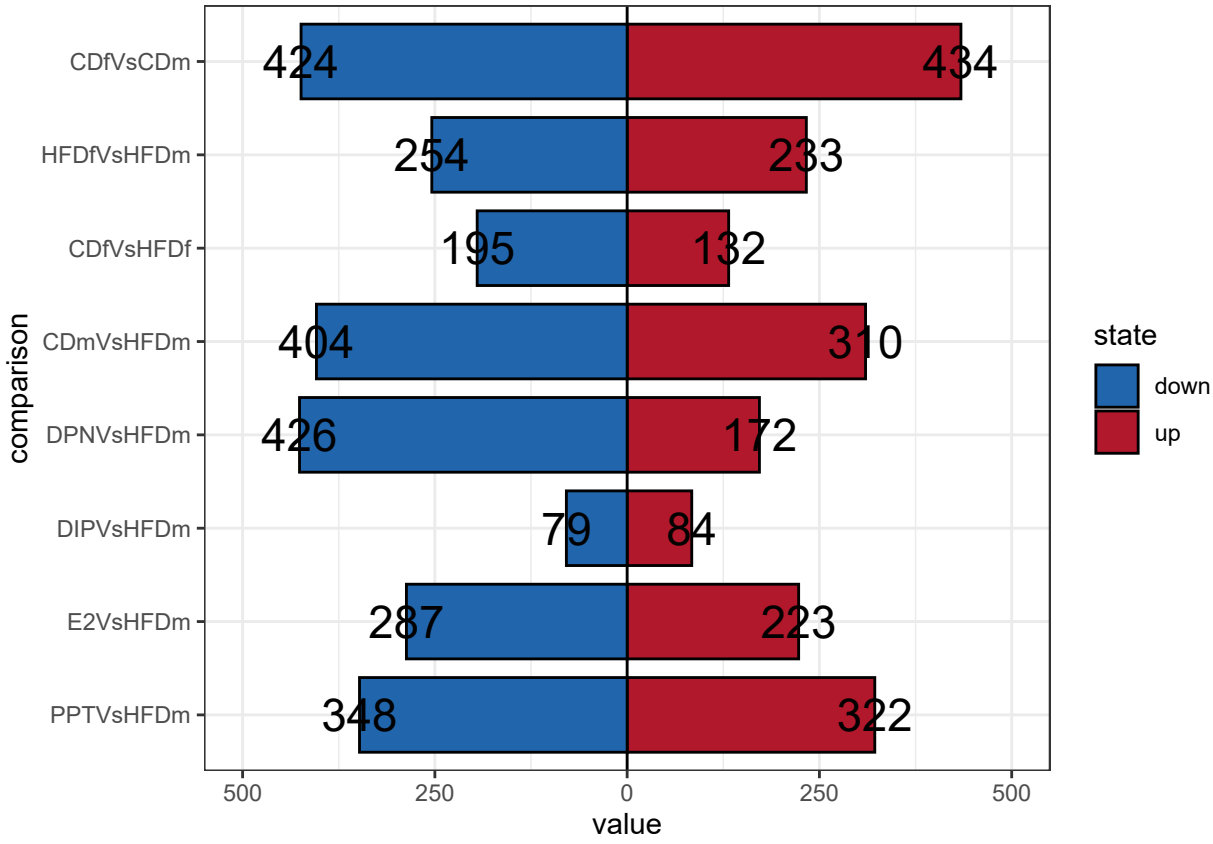
```
pca_res <- DESeq2::DESeqDataSetFromMatrix(countData = raw_counts,
                                  colData = design_meta,
                                  design = ~0 + condition) %>%
  DESeq2::estimateSizeFactors() %>%
  DESeq2::DESeq() %>%
  DESeq2::vst(blind = FALSE) %>%
  assay() %>%
  doPCA()

df <- data.frame(PC1 = pca_res$pcs$PC1,
                 PC2 = pca_res$pcs$PC2,
                 condition = design_meta$condition)

ggplot(df, aes(x=PC1, y=PC2, fill=condition),) +
  geom_point(shape=21, size=5, stroke=0.5, color='black') +
  scale_fill_manual(values = alpha(colPals$conditions, 0.9)) +
  scale_x_continuous(expand = expansion(mult = c(.1, .1))) +
  scale_y_continuous(
    expand =expansion(mult = c(.1, .1))) +
  xlab(paste0('PC1 (', round(pca_res$perc_var[1]), '%)')) +
  ylab(paste0('PC2 (', round(pca_res$perc_var[2]), '%)')) +
  theme_bw()
```



Differentially expressed genes (DEGs)

```
up <- lapply(DEGs$filt, function(x) sum(x$log2FoldChange>0)) %>% unlist()
down <- lapply(DEGs$filt, function(x) sum(x$log2FoldChange<0)) %>% unlist()

df <- data.frame(comparison=factor(rep(names(DEGs$filt),2), levels=names(DEGs$filt)),
                 state=c(rep('up', length(up)), rep('down', length(down))),
                 value=c(up, down*-1))
```

```
ggplot(df, aes(x=comparison, y=value, fill=state, label=abs(value))) +
  geom_hline(yintercept = 0, linetype='solid', size=0.5) +
  geom_bar(color='black', size=0.5, width=0.8, position='stack', stat='identity') +
  geom_text(size=6) +
  scale_fill_manual(values = colPals$UpDown) +
  scale_x_discrete(limits = rev) +
  scale_y_continuous(limits = c(-500,500), labels = c(500,250,0,250,500)) +
  coord_flip() +
  theme_bw()
```

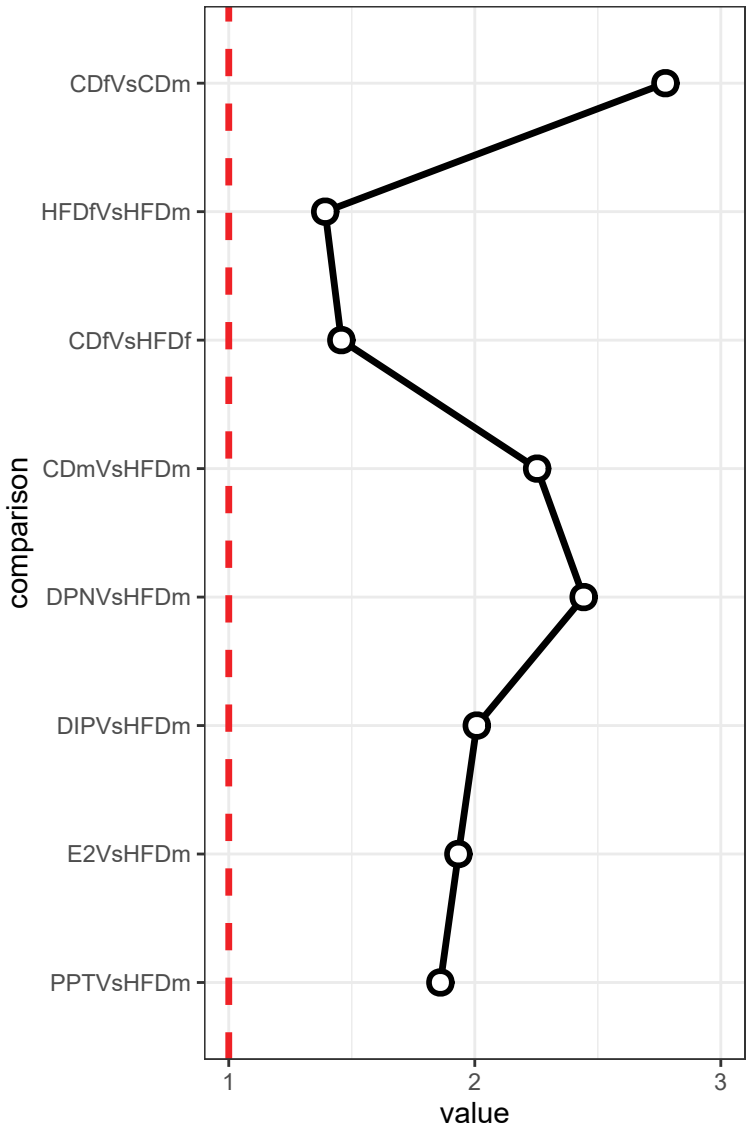


Filter and normalize RNAseq data

```
RNAseq <- list()

# remove outlier sample PPT_HFD_male_4
RNAseq$counts <- raw_counts %>%
  as.data.frame() %>%
  dplyr::select(-PPT_HFD_male_4)

RNAseq$design_meta <- design_meta %>%
  dplyr::filter(sample != 'PPT_HFD_male_4')

RNAseq$annotation <- gene_ann %>%
  dplyr::rename(geneID = ensembl_gene_id) %>%
  dplyr::left_join(gene_len, by = 'geneID')

RNAseq$cpm <- RNAseq$counts %>%
  normalizeData(method = 'CPM')

RNAseq$tpm <- RNAseq$counts %>%
  normalizeData(len = RNAseq$annotation$length, method = 'TPM')
```

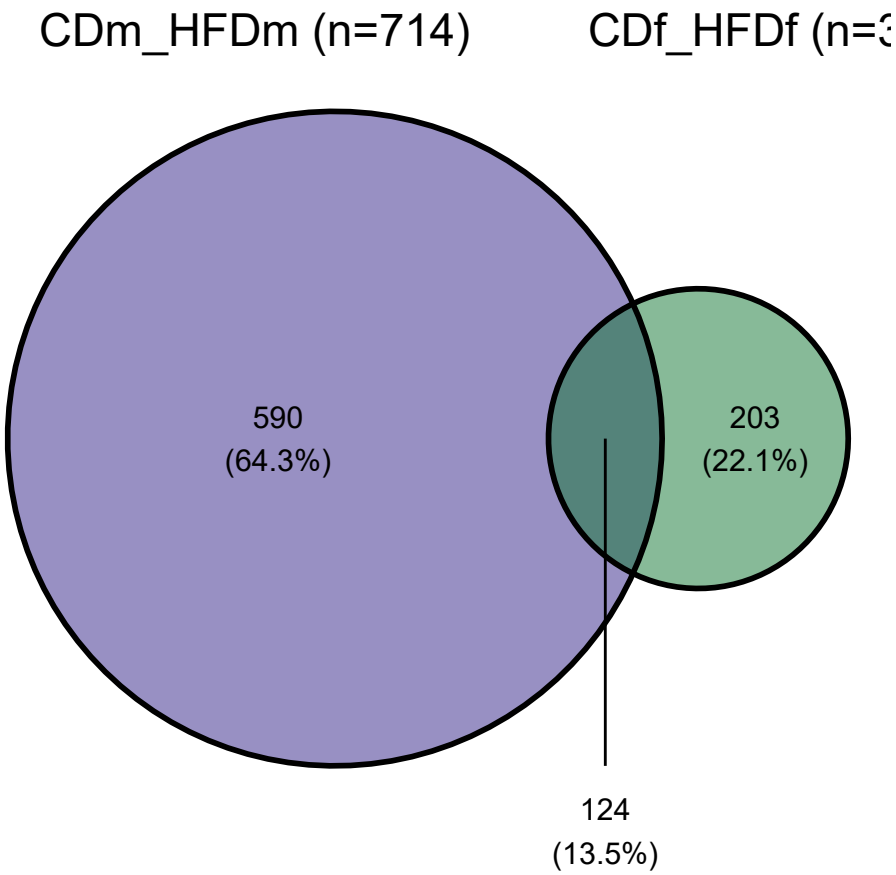
Transcriptome-wide signal-to-noise ratios (tSNR)

```
df <- RNAseq$tpm %>%
  scaleData(method = 'zscore') %>%
  tSNR(group.lbls = RNAseq$design_meta$condition) %>%
  tibble::rownames_to_column(var = 'X') %>%
  tidyr::pivot_longer(cols = dplyr::everything()[-1], names_to = 'Y') %>%
  tidyr::unite(col = 'comparison', X, Y, sep = 'Vs') %>%
  dplyr::filter(comparison %in% names(DEGs$filt)) %>%
  dplyr::mutate(comparison=factor(comparison, levels = names(DEGs$filt)))

ggplot(df, aes(x=comparison, y=value)) +
  geom_line(group=1, size=1.2) +
  geom_point(shape=21, size=3, stroke=1.5, color='black', fill='white') +
  geom_hline(yintercept = 1, linetype='dashed', size=1.2, color='#EF2126') +
  scale_x_discrete(limits = rev) +
  scale_y_continuous(limits = c(1,3), breaks = c(1,2,3)) +
  coord_flip() +
  theme_bw()
```



Plot Venn Diagram (Fig S1E)

```
CDf_HFDf <- DEGs$filt$CDfVsHFDf
CDm_HFDm <- DEGs$filt$CDmVsHFDm

# Create Data
venn <- list("CDf_HFDf (n=327)" = sort(CDf_HFDf$ensembl_gene_id),
                  "CDm_HFDm (n=714)"= CDm_HFDm$ensembl_gene_id)

# Create venn diagram Pairwise
ggvenn(venn, c("CDm_HFDm (n=714)", "CDf_HFDf (n=327)"),
       auto_scale=TRUE,
       fill_color = c("#332288", "#117733"))
```

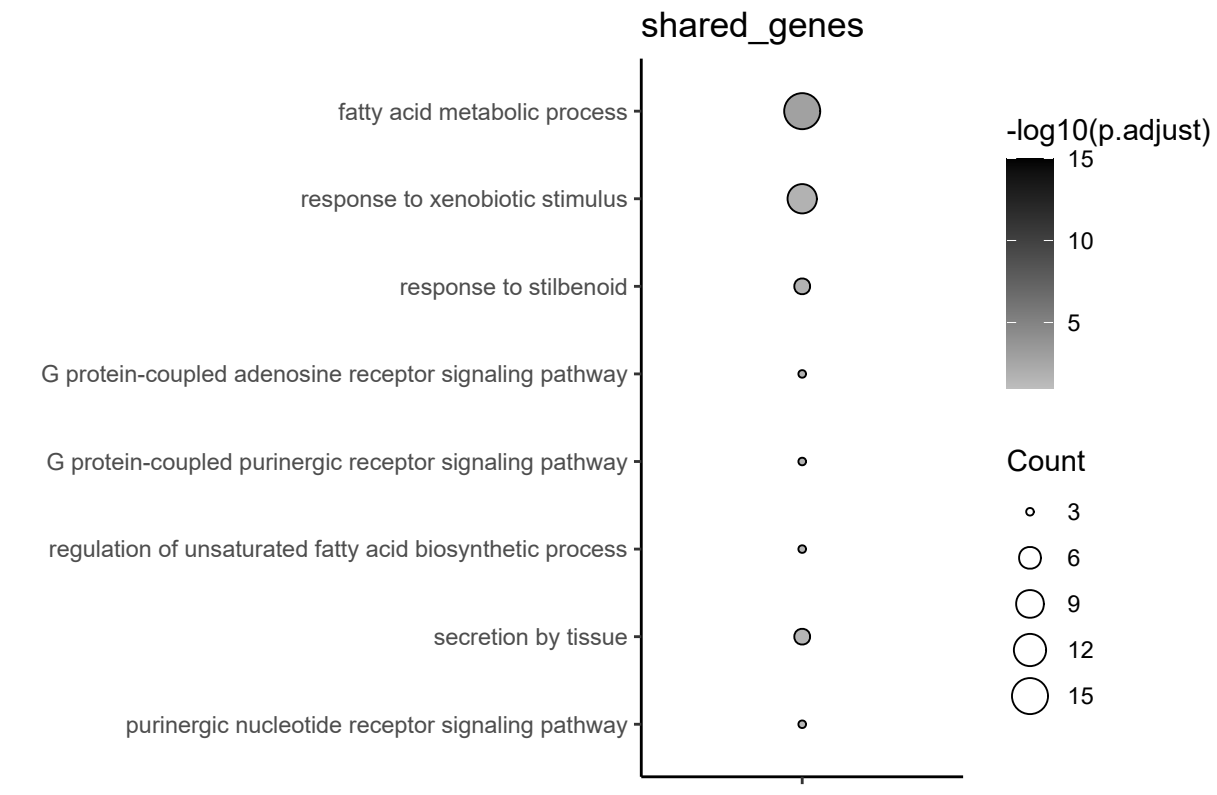


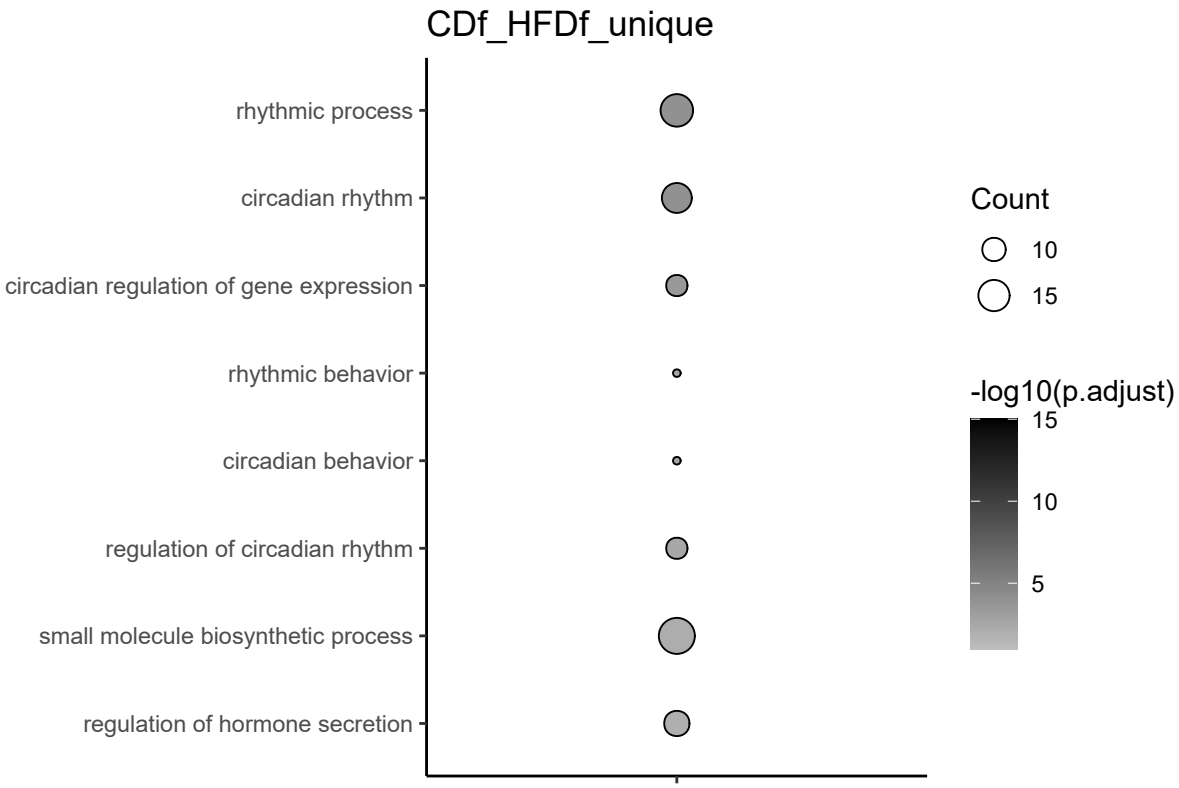

```
# Make intersection to extract the genes names from Venn.
intersections <- list("", "", "")
names(intersections) <- c("shared_genes", "CDf_HFDf_unique", "CDm_HFDm_unique")

intersections[[1]] <- intersect(CDf_HFDf$ensembl_gene_id,CDm_HFDm$ensembl_gene_id)
intersections[[2]] <- setdiff(CDf_HFDf$ensembl_gene_id, intersections$shared_genes)
intersections[[3]] <- setdiff(CDm_HFDm$ensembl_gene_id, intersections$shared_genes)
```

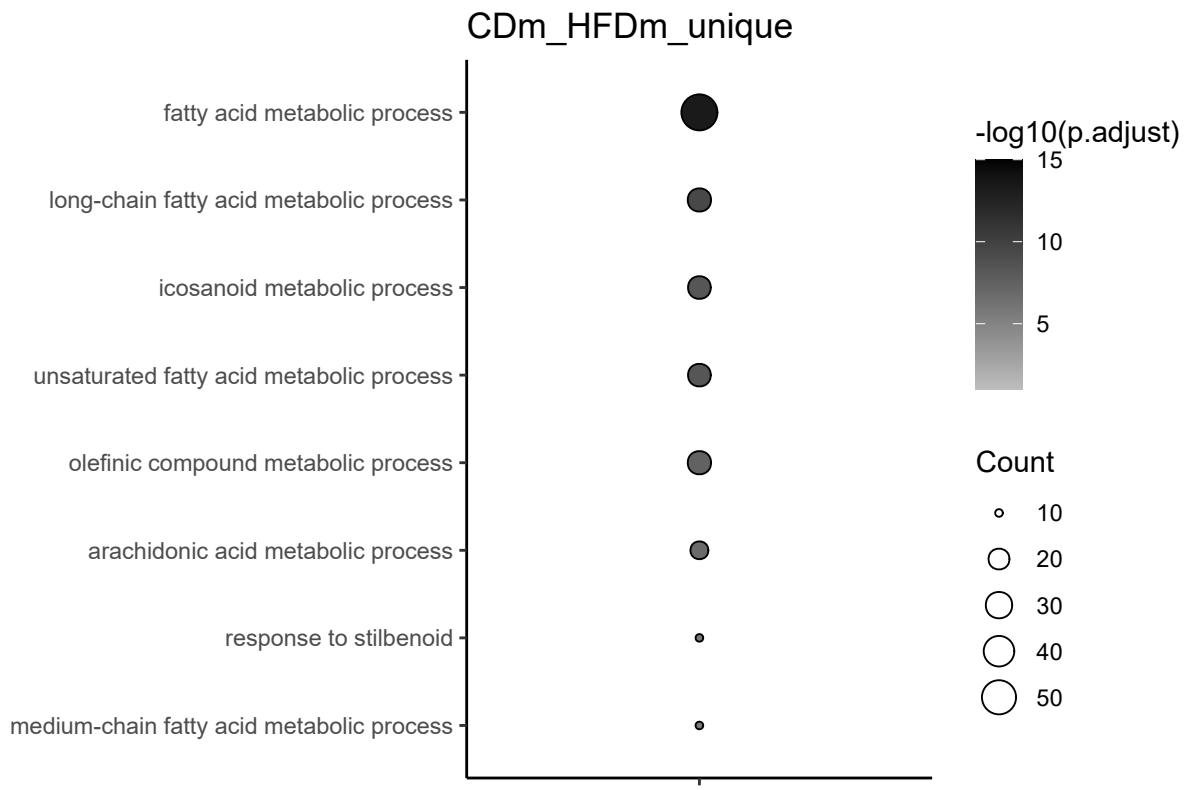
Gene Ontologies (Fig S1E)

```
background <- DEGs$unfilt$CDfVsCDm$ensembl_gene_id
options(connectionObserver = NULL) # workaround due to a bug

GO_list <- list(GO.results = list(),
                GO.top8 = list(),
                term.order.plotting = list())

for (i in 1:3) {
  GO_list$GO.results[[i]] <- enrichGO(gene = intersections[[i]],
                                      keyType       = 'ENSEMBL',
                                      OrgDb         = org.Mm.eg.db,
```

```
                                    ont           = "BP",
                                    pAdjustMethod = "BH",
                                    pvalueCutoff  = 0.05,
                                    qvalueCutoff  = 0.05,
                                    minGSSize     = 3,
                                    readable      = TRUE,
                                    universe = background)
  head(GO_list$GO.results[[i]])

  name.me <- c("shared_genes", "CDf_HFDf_unique", "CDm_HFDm_unique")


  GO_list$GO.top8[[i]] <- GO_list$GO.results[[i]]@result %>% filter(p.adjust<0.05) %>% mutate(GeneSet = name.me[i]) %>% dplyr::arrange(p.adjust) %

  GO_list$term.order.plotting[[i]] <- GO_list$GO.top8[[i]] %>% dplyr::pull("Description")

  print(ggplot(GO_list$GO.top8[[i]], aes(x=factor(""), y=factor(Description, levels=rev(GO_list$term.order.plotting[[i]])))) +
          geom_point(shape=21, aes(size=Count, fill=-log10(p.adjust))) +
          theme_classic() +
          xlab("") +
          ylab("") +
          ggtitle(paste(unique(GO_list$GO.top8[[i]]$GeneSet))) +
          scale_fill_gradient(low="grey", high= "black", limits=c(1,15)))

  # write.table(GO_list$GO.results[[i]]@result, paste0("Supplementary_tables/FigS1E_GO_", unique(GO_list$GO.top8[[i]]$GeneSet), ".txt"), sep="\t", q
}
```




CDf_HFDf_unique
rhythmic process
circadian rhythm
circadian regulation of gene expression
rhythmic behavior
circadian behavior
regulation of circadian rhythm
small molecule biosynthetic process
regulation of hormone secretion
Count
10
15
-log10(p.adjust)
15
10
5



Export filtered and normalized RNAseq data

```
saveRDS(RNAseq, file = 'results/bulkRNAseq_mmus_data_filt_norm.rds')
```

SessionInfo

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## attached base packages:
## [1] grid      stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] ggrepel_0.9.2               org.Mm.eg.db_3.16.0
##  [3] AnnotationDbi_1.60.2        clusterProfiler_4.6.2
##  [5] ggvenn_0.1.10               DESeq2_1.38.1
##  [7] SummarizedExperiment_1.28.0 Biobase_2.58.0
##  [9] MatrixGenerics_1.10.0       matrixStats_0.63.0
## [11] GenomicRanges_1.50.1        GenomeInfoDb_1.34.9
## [13] IRanges_2.32.0              S4Vectors_0.36.0
```

```
## [15] BiocGenerics_0.44.0         lubridate_1.9.2
## [17] forcats_1.0.0               stringr_1.5.0
## [19] dplyr_1.1.0                 purrr_1.0.1
## [21] readr_2.1.4                 tidyr_1.3.0
## [23] tibble_3.2.1                ggplot2_3.4.2
## [25] tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##   [1] ggtree_3.6.2           fgsea_1.24.0           colorspace_2.0-3
##   [4] gson_0.1.0             qvalue_2.30.0          XVector_0.38.0
##   [7] aplot_0.1.10           rstudioapi_0.15.0      farver_2.1.1
##  [10] graphlayouts_0.8.4     bit64_4.0.5            scatterpie_0.2.1
##  [13] fansi_1.0.3            codetools_0.2-19       splines_4.2.1
##  [16] cachem_1.0.6           GOSemSim_2.24.0        geneplotter_1.76.0
##  [19] knitr_1.43             polyclip_1.10-4        jsonlite_1.7.2
##  [22] annotate_1.76.0        GO.db_3.16.0           png_0.1-8
##  [25] ggforce_0.4.1          compiler_4.2.1         httr_1.4.6
##  [28] lazyeval_0.2.2         Matrix_1.5-3           fastmap_1.1.0
##  [31] cli_3.4.1              tweenr_2.0.2           htmltools_0.5.5
##  [34] tools_4.2.1            igraph_1.3.5           gtable_0.3.3
##  [37] glue_1.6.2             GenomeInfoDbData_1.2.9 reshape2_1.4.4
##  [40] fastmatch_1.1-3        Rcpp_1.0.9             enrichplot_1.18.4
##  [43] vctrs_0.6.2            Biostrings_2.66.0      nlme_3.1-162
##  [46] ape_5.6-2              ggraph_2.1.0           xfun_0.39
##  [49] timechange_0.2.0       lifecycle_1.0.3        XML_3.99-0.12
##  [52] DOSE_3.24.2            zlibbioc_1.44.0        MASS_7.3-60
##  [55] scales_1.2.1           tidygraph_1.2.2        hms_1.1.3
##  [58] parallel_4.2.1         RColorBrewer_1.1-3     yaml_2.3.7
##  [61] memoise_2.0.1          gridExtra_2.3          downloader_0.4
##  [64] ggfun_0.1.1            HDO.db_0.99.1          yulab.utils_0.0.6
##  [67] stringi_1.7.8          RSQLite_2.2.19         highr_0.10
##  [70] tidytree_0.4.4         BiocParallel_1.32.3    rlang_1.1.1
##  [73] pkgconfig_2.0.3        bitops_1.0-7           evaluate_0.21
##  [76] lattice_0.20-41        labeling_0.4.2         treeio_1.22.0
##  [79] patchwork_1.1.2        shadowtext_0.1.2       cowplot_1.1.1
##  [82] bit_4.0.5              tidyselect_1.2.0       plyr_1.8.8
##  [85] magrittr_2.0.3         R6_2.5.1               generics_0.1.3
##  [88] DelayedArray_0.24.0    DBI_1.1.3              pillar_1.9.0
##  [91] withr_2.5.0            KEGGREST_1.38.0        RCurl_1.98-1.9
##  [94] crayon_1.5.2           utf8_1.2.2             tzdb_0.4.0
##  [97] rmarkdown_2.23         viridis_0.6.3          locfit_1.5-9.6
## [100] data.table_1.14.6      blob_1.2.4             digest_0.6.30
## [103] xtable_1.8-4           gridGraphics_0.5-1     munsell_0.5.0
## [106] viridisLite_0.4.2      ggplotify_0.1.1
```